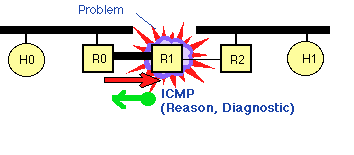
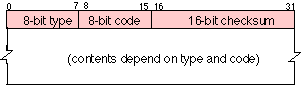
**The Internet Control Message Protocol (ICMP)**

The Internet Control Message Protocol (ICMP) [RFC792] protocol is classic example of a client server application. The ICMP server executes on all IP end system computers and all IP intermediate systems (i.e routers). The protocol is used to report problems with delivery of IP datagrams within an IP network. It can be sued to show when a particular End System (ES) is not responding, when an IP network is not reachable, when a node is overloaded, when an error occurs in the IP header information, etc. The protocol is also frequently used by Internet managers to verify correct operations of End Systems (ES) and to check that routers are correctly routing packets to the specified destination address.



ICMP messages generated by router R1, in response to message sent by H0 to H1 and forwarded by R0. This message could, for instance be generated if the MTU of the link between R0 and R1 was smaller than size of the IP packet, and the packet had the Don't Fragment (DF) bit set in the IP packet header. The ICMP message is returned to H0, since this is the source address specified in the IP packet that suffered the problem. A modern version of Path MTU Discovery provides a mechanism to verify the Path MTU [RFC4821].



An ICMP message consisting of 4 bytes of PCI and an optional message payload.

The format of an ICMP message is shown above. The 8-bit type code identifies the types of message. This is followed by at least the first 28 bytes of the packet that resulted in generation of the error message (i.e. the network-layer header and first 8 bytes of transport header). This payload is, for instance used by a sender that receives the ICMP message to perform Path MTU Discoveryso that it may determine IP destination address of the packet that resulted in the error. Longer payloads are also encouraged (which can help better identify the reason why the ICMP message was generated and which program generated the original packet).

The figure below shows the encapsulation of ICMP over an Ethernet LAN using an IP network layer header, and a MAC link layer header and trailer containing the 32-bit checksum:

http://www.erg.abdn.ac.uk/users/gorry/eg3567/images/icmp-encap.gif

Encapsulation for a complete ICMP packet (not showing the Ethernet preamble)

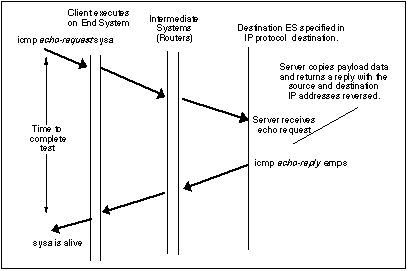
It is the responsibility of the network layer (IP) protocol to ensure that the ICMP message is sent to the correct destination. This is achieved by setting the destination address of the IP packet carrying the ICMP message. The source address is set to the address of the computer that generated the IP packet (carried in the IP source address field) and the IP protocol type is set to "ICMP" to indicate that the packet is to be handled by the remote end system's ICMP client interface.

RFC792 specifies the Internet Control Message Protocol (ICMP) that is used with the Internet Protocol version 4 (IPv4). It defines, among other things, a number of error messages that can be used by an end-system and intermediate systems to report errors back to the sending system. The host requirements [RFC1122] classifies ICMP these error messages into those that indicate "soft errors" (advising of problems), and those that indicate "hard errors" (which need to be responded to).

A version of ICMP has also been defined for IPv6, called ICMPv6 [RFC4443]. This subsumes all the equivalent functions of ICMP for IPv4 and adds other network-layer functions. ICMP error messages are up to 1280 bytes in size, and therefore always carry a substantial number of bytes from the packet that generated the error being reported.

The Ping Application

The "ping" program contains a client interface to ICMP. It may be used by a user to verify an end-to-end Internet Path is operational. The ping program also collects performance statistics (i.e. the measured round trip time and the number of times the remote server fails to reply. Each time an ICMP echo reply message is received, the ping program displays a single line of text. The text printed by ping shows the received sequence number, and the measured round trip time (in milliseconds). Each ICMP Echo message contains a sequence number (starting at 0) that is incremented after each transmission, and a timestamp value indicating the transmission time.



Use of the ping program to test whether a particular computer ("sysa") is operational.

The operation of ICMP is illustrated in the frame transition diagram shown above. In this case there is only one Intermediate System (IS) (i.e. IP router). In this case two types of message are involved

the ECHO request (sent by the client) and the ECHO reply (the response by the server). Each message may contain some optional data. When data are sent by a server, the server returns the data in the reply which is generated. ICMP packets are encapsulated in IP for transmission across an internet.

The Traceroute Application

The "traceroute" program contains a client interface to ICMP. Like the "ping" program, it may be used by a user to verify an end-to-end Internet Path is operational, but also provides information on each of the Intermediate Systems (i.e. IP routers) to be found along the IP Path from the sender to the receiver. Traceroute uses ICMP echo messages. These are addressed to the target IP address. The sender manipulates the TTL (hop count) value at the IP layer to force each hop in turn to return an error message.

* The program starts by sending an ICMP Echo request message with an IP destination address of the system to be tested and with a Time To Live (TTL) value set to 1. The first router that receives this packet decrements the TTL and discards the message, since this now has a value of zero. Before it deletes the message, the system constructs an ICMP error message (with an ICMP message type of "TTL exceeded") and returns this back to the sender. Receipt of this message allows the sender to identify which system is one link away along the path to the specified destination.
* The sender repeats this two more times, each time reporting the system that received the packet. If all packets travel along the same path, each ICMP error message will be received from the same system. Where two or more alternate paths are being used, the results may vary.
* If the system that responded was not the intended destination, the sender repeats the process by sending a set of three identical messages, but using a TTL value that is one larger than the previous attempt. The first system forwards the packet (decrementing the TTL value in the IP header), but a subsequent system that reduces the TTL value to zero, generates an ICMP error message with its own source address. In this way, the sender learns the identity of another system along the IP path to the destination.
* This process repeats until the sender receives a response from the intended destination (or the maximum TTL value is reached).

# Address Resolution Protocol (arp)

The address resolution protocol (arp) is a protocol used by the Internet Protocol (IP) [RFC826], specifically IPv4, to map IP network addresses to the hardware addresses used by a data link protocol. The protocol operates below the network layer as a part of the interface between the OSI network and OSI link layer. It is used when IPv4 is used over Ethernet.

The term address resolution refers to the process of finding an address of a computer in a network. The address is "resolved" using a protocol in which a piece of information is sent by a client process executing on the local computer to a server process executing on a remote computer. The information received by the server allows the server to uniquely identify the network system for which the address was required and therefore to provide the required address. The address resolution procedure is completed when the client receives a response from the server containing the required address.

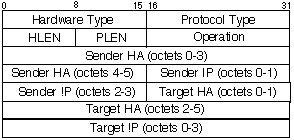
An Ethernet network uses two hardware addresses which identify the source and destination of each frame sent by the Ethernet. The destination address (all 1's) may also identify a broadcastpacket (to be sent to all connected computers). The hardware address is also known as the Medium Access Control (MAC) address, in reference to the standards which define Ethernet. Each computer network interface card is allocated a globally unique 6 byte link address when the factory manufactures the card (stored in a PROM). This is the normal link source address used by an interface. A computer sends all packets which it creates with its own hardware source link address, and receives all packets which match the same hardware address in the destination field or one (or more) pre-selected broadcast/multicast addresses.

The Ethernet address is a link layer address and is dependent on the interface card which is used. IP operates at the network layer and is not concerned with the link addresses of individual nodes which are to be used.The address resolution protocol (arp) is therefore used to translate between the two types of address. The arp client and server processes operate on all computers using IP over Ethernet. The processes are normally implemented as part of the software driver that drives the network interface card.

There are four types of arp messages that may be sent by the arp protocol. These are identified by four values in the "operation" field of an arp message. The types of message are:

1. ARP request
2. ARP reply
3. RARP request
4. RARP reply

The format of an arp message is shown below:



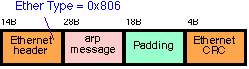
Format of an arp message used to resolve the remote MAC Hardware Address (HA)

To reduce the number of address resolution requests, a client normally caches resolved addresses for a (short) period of time. The arp cache is of a finite size, and would become full of incomplete and obsolete entries for computers that are not in use if it was allowed to grow without check. The arp cache is therefore periodically flushed of all entries. This deletes unused entries and frees space in the cache. It also removes any unsuccessful attempts to contact computers which are not currently running.

If a host changes the MAC address it is using, this can be detected by other hosts when the cache entry is deleted and a fresh arp message is sent to establish the new association. The use of gratuitous arp (e.g. triggered when the new NIC interface is enabled with an IP address) provides a more rapid update of this information.

### Example of use of the Address Resolution Protocol (arp)

The figure below shows the use of arp when a computer tries to contact a remote computer on the same LAN (known as "sysa") using the "ping" program. It is assumed that no previous IP datagrams have been received form this computer, and therefore arp must first be used to identify the MAC address of the remote computer.



The arp request message ("who is X.X.X.X tell Y.Y.Y.Y", where X.X.X.X and Y.Y.Y.Y are IP addresses) is sent using the Ethernet broadcast address, and an Ethernet protocol type of value 0x806. Since it is broadcast, it is received by all systems in the same collision domain (LAN). This is ensures that

is the target of the query is connected to the network, it will receive a copy of the query. Only this system responds. The other systems discard the packet silently.

The target system forms an arp response ("X.X.X.X is hh:hh:hh:hh:hh:hh", where hh:hh:hh:hh:hh:hh is the Ethernet source address of the computer with the IP address of X.X.X.X). This packet is unicast to the address of the computer sending the query (in this case Y.Y.Y.Y). Since the original request also included the hardware address (Ethernet source address) of the requesting computer, this is already known, and doesn't require another arp message to find this out.

### http://www.erg.abdn.ac.uk/users/gorry/course/images/arp-eg.gif

Gratuitous ARP

Gratuitous ARP is used when a node (end system) has selected an IP address and then wishes to defend its chosen address on the local area network (i.e. to check no other node is using the same IP address). It can also be used to force a common view of the node's IP address (e.g. after the IP address has changed).

Use of this is common when an interface is first configured, as the node attempts to clear out any stale caches that might be present on other hosts. The node simply sends an arp request for itself.

Proxy ARP

Proxy ARP is the name given when a node responds to an arp request on behalf of another node. This is commonly used to redirect traffic sent to one IP address to another system.

Proxy ARP can also be used to subvert traffic away from the intended recipient. By responding instead of the intended recipient, a node can pretend to be a different node in a network, and therefore force traffic

directed to the node to be redirected to itself. The node can then view the traffic (e.g. before forwarding this to the originally intended node) or could modify the traffic. Improper use of Proxy ARP is therefore a significant security vulnerability and some networks therefore implement systems to detect this. Gratuitous ARP can also help defend the correct IP to MAC bindings.

**The Reverse Address Resolution Protocol (RARP)**

The first method devised to address the bootstrapping problem in TCP/IP was the backwards use of ARP. This technique was formalized in RFC 903, A Reverse Address Resolution Protocol (RARP), published in 1984. Where ARP allows device A to say “I am device A and I have device B's IP address, device B please tell me your hardware address”, RARP is used by device A to say “I am device A and I am sending this broadcast using my hardware address, can someone please tell me my IP address?”.The two-step operation of RARP is illustrated in Figure.

|  |
| --- |
| http://www.tcpipguide.com/free/diagrams/rarptrans.png |
| Figure: Operation of the Reverse Address Resolution Protocol (RARP)  RARP, as the name suggests, works like ARP but in reverse, instead of Device A providing the IP address of another device and asking for its hardware address, it is providing its own hardware address and asking for an IP address it can use. The answer, in this case, is provided by Device D, which is serving as an RARP server for this network. |

RARP is not only very similar to ARP, it basically is ARP. What I mean by this is that RFC 903 doesn't define a whole new protocol from scratch, it just describes a new method for using ARP to perform the opposite of its normal function. RARP uses ARP messages in exactly the same format as ARP, but uses different opcodes to accomplish its reverse function. Just as in ARP, a request and reply are used in an exchange. The meaning of the address fields is the same too: the sender is the device transmitting a message while the target is the one receiving it.

RARP General Operation

Here are the steps followed in a RARP transaction (illustrated in Figure 53):

|  |
| --- |
| http://www.tcpipguide.com/free/diagrams/rarpoperation.png |
| Figure 53: Reverse Address Resolution Protocol (RARP) Operation  RARP uses a simple request/reply exchange to allow a device to obtain an IP address. |

1. Source Device Generates RARP Request Message: The source device generates an RARP Request message. Thus, it uses the value 3 for theOpcode in the message. It puts its own data link layer address as both the Sender Hardware Address and also the Target Hardware Address. It leaves both the Sender Protocol Address and the Target Protocol Address blank, since it doesn't know either.
2. Source Device Broadcasts RARP Request Message: The source broadcasts the ARP Request message on the local network.
3. Local Devices Process RARP Request Message: The message is received by each device on the local network and processed. Devices that are not configured to act as RARP servers ignore the message.
4. RARP Server Generates RARP Reply Message: Any device on the network that is set up to act as an RARP server responds to the broadcast from the source device. It generates an RARP Reply using an Opcode value of 4. It sets the Sender Hardware Address and Sender Protocol Address to its own hardware and IP address of course, since it is the sender of the reply. It then sets the Target Hardware Address to the hardware address of the original source device. It looks up in a table the hardware address of the source, determines that device's IP address assignment, and puts it into the Target Protocol Address field.
5. RARP Server Sends RARP Reply Message: The RARP server sends the RARP Reply message unicast to the device looking to be configured.
6. Source Device Processes RARP Reply Message: The source device processes the reply from the RARP server. It then configures itself using the IP address in the Target Protocol Address supplied by the RARP server.

It is possible that more than one RARP server may respond to any request, if two or more are configured on any local network. The source device will typically use the first reply and discard the others.

Limitations of RARP

RARP is the earliest and most rudimentary of the class of technologies I call host configuration protocols, which I describe in general terms in a dedicated section. As the first of these protocols, RARP was a useful addition to the TCP/IP protocol in the early 1980s, but has several shortcomings, the most important of which are:

* Low-Level Hardware Orientation: RARP works using hardware broadcasts. This means that if you have a large internetwork with many physical networks, you need an RARP server on every network segment. Worse, if you need reliability to make sure RARP keeps running even

if one RARP server goes down, you need two on each physical network. This makes centralized management of IP addresses difficult.

* Manual Assignment: RARP allows hosts to configure themselves automatically, but the RARP server must still be set up with a manual table of bindings between hardware and IP addresses. These must be maintained for each server, which is again a lot of work on an administrator.
* Limited Information: RARP only provides a host with its IP address. It cannot provide other needed information such as, for example, a subnet mask or default gateway.

Today, the importance of host configuration has increased since the early 1980s. Many organizations assign IP addresses dynamically even for hosts that have disk storage, because of the many advantages this provides in administration and efficient use of address space. For this reason, RARP has been replaced by two more capable technologies that operate at higher layers in the TCP/IP protocol stack: BOOTP and DHCP. They are discussed inthe application layer section on host configuration protocols.

### **Internet Group Management Protocol (IGMP)**

IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN. Hosts identify group memberships by sending IGMP messages to their local multicast router. Under IGMP, routers listen to IGMP messages and periodically send out queries to discover which groups are active or inactive on a particular subnet.

IGMP versions are described in the following sections.

### IGMP Version 1

RFC 1112, Host Extensions for IP Multicasting, describes the specification for IGMP Version 1 (IGMPv1). A diagram of the packet format for an IGMPv1 message is shown in [Figure 5](http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html#wp1008786).

Figure 5 IGMPv1 Message Format

http://www.cisco.com/c/dam/en/us/td/i/000001-100000/60001-65000/60001-61000/60075.ps/_jcr_content/renditions/60075.jpg

In Version 1, only the following two types of IGMP messages exist:

•http://www.cisco.com/c/dam/en/us/td/i/templates/blank.gifMembership query

•http://www.cisco.com/c/dam/en/us/td/i/templates/blank.gifMembership report

Hosts send out IGMP membership reports corresponding to a particular multicast group to indicate that they are interested in joining that group. The TCP/IP stack running on a host automatically sends the IGMP Membership report when an application opens a multicast socket. The router periodically sends out an IGMP membership query to verify that at least one host on the subnet is still interested in receiving traffic directed to that group. When there is no reply to three consecutive IGMP membership queries, the router times out the group and stops forwarding traffic directed toward that group.

### IGMP Version 2

IGMPv1 has been superceded by IGMP Version 2 (IGMPv2), which is now the current standard. IGMPv2 is backward compatible with IGMPv1. RFC 2236, Internet Group Management Protocol, Version 2, describes the specification for IGMPv2. A diagram of the packet format for an IGMPv2 message is shown in Figure.

Figure  IGMPv2 Message Format

http://www.cisco.com/c/dam/en/us/td/i/000001-100000/60001-65000/60001-61000/60076.ps/_jcr_content/renditions/60076.jpg

In Version 2, the following four types of IGMP messages exist:

•http://www.cisco.com/c/dam/en/us/td/i/templates/blank.gifMembership query

•http://www.cisco.com/c/dam/en/us/td/i/templates/blank.gifVersion 1 membership report

•http://www.cisco.com/c/dam/en/us/td/i/templates/blank.gifVersion 2 membership report

•http://www.cisco.com/c/dam/en/us/td/i/templates/blank.gifLeave group

IGMP Version 2 works basically the same way as Version 1. The main difference is that there is a leave group message. With this message, the hosts can actively communicate to the local multicast router that they intend to leave the group. The router then sends out a group-specific query and determines if any remaining hosts are interested in receiving the traffic. If there are no replies, the router times out the group and stops forwarding the traffic. The addition of the leave group message in IGMP Version 2 greatly reduces the leave latency compared to IGMP Version 1. Unwanted and unnecessary traffic can be stopped much sooner.

### 

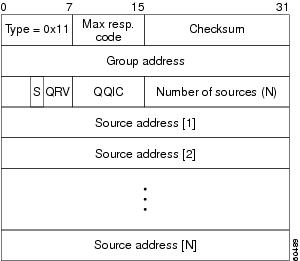
### IGMP Version 3

IGMP Version 3 (IGMPv3) is the next step in the evolution of IGMP. IGMPv3 adds support for "source filtering," which enables a multicast receiver host to signal to a router the groups from which it wants to receive multicast traffic, and from which sources this traffic is expected. This membership information enables Cisco IOS software to forward traffic from only those sources from which receivers requested the traffic.

IGMPv3 is an emerging standard. The latest versions of Windows, Macintosh, and UNIX operating systems all support IGMPv3. At the time this document was being written, application developers were in the process of porting their applications to the IGMPv3 API.

A diagram of the query packet format for an IGMPv3 message is shown in [Figure 7](http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html#wp1008825).

Figure 7 IGMPv3 Query Message Format

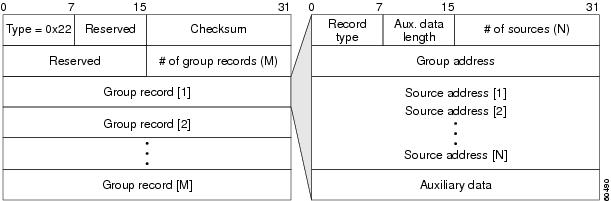


[Table 3](http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html#wp1008836) describes the significant fields in an IGMPv3 query message.

|  |  |
| --- | --- |
| Table 3 IGMPv3 Query Message Field Descriptions | |
| Field | Description |
| Type = 0x11 | IGMP query. |
| Max resp. code | Maximum response code (in seconds). This field specifies the maximum time allowed before sending a responding report. |
| Group address | Multicast group address. This address is 0.0.0.0 for general queries. |
| S | S flag. This flag indicates that processing by routers is being suppressed. |
| QRV | Querier Robustness Value. This value affects timers and the number of retries. |
| QQIC | Querier's Query Interval Code (in seconds). This field specifies the Query Interval used by the querier. |
| Number of sources [N] | Number of sources present in the query. This number is nonzero for a group-and-source query. |
| Source address [1...N] | Address of the source(s). |

A diagram of the report packet format for an IGMPv3 message is shown in [Figure 8](http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html#wp1008884).

Figure 8 IGMPv3 Report Message Format



[Table 4](http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html#wp1008895) describes the significant fields in an IGMPv3 report message.

|  |  |
| --- | --- |
| Table 4 IGMPv3 Report Message Field Descriptions | |
| Field | Description |
| # of group records [M] | Number of group records present in the report. |
| Group record [1...M] | Block of fields containing information regarding the sender's membership with a single multicast group on the interface from which the report was sent. |
| Record type | The group record type (e.g., MODE\_IS\_INCLUDE, MODE\_IS\_EXCLUDE). |
| # of sources [N] | Number of sources present in the record. |
| Source address [1...N] | Address of the source(s). |

In IGMPv3, the following types of IGMP messages exist:

•http://www.cisco.com/c/dam/en/us/td/i/templates/blank.gifVersion 3 membership query

•http://www.cisco.com/c/dam/en/us/td/i/templates/blank.gifVersion 3 membership report

IGMPv3 supports applications that explicitly signal sources from which they want to receive traffic. With IGMPv3, receivers signal membership to a multicast host group in the following two modes:

•http://www.cisco.com/c/dam/en/us/td/i/templates/blank.gifINCLUDE mode—In this mode, the receiver announces membership to a host group and provides a list of source addresses (the INCLUDE list) from which it wants to receive traffic.

•http://www.cisco.com/c/dam/en/us/td/i/templates/blank.gifEXCLUDE mode—In this mode, the receiver announces membership to a multicast group and provides a list of source addresses (the EXCLUDE list) from which it does not want to receive traffic. The host will receive traffic only from sources whose IP addresses are not listed in the EXCLUDE list. To receive traffic from all sources, which is the behavior of IGMPv2, a host uses EXCLUDE mode membership with an empty EXCLUDE list.